Final Report on Solving Large Computational Problems on Parallel Computers N00014-88-K-0563 July 1, 1988 to August 31, 1989

Submitted to

Office of Naval Research

by

**School of Computer Science
Carnegie Mellon University
Pittsburgh, Pa. 15213**

1. Introduction

Shared memory parallel machines are relatively easy to program, but it is difficult to scale up their performance by increasing the number of processors because the bandwidth of the shared memory is limited. Although distributed memory parallel computers are more difficult to program, they have the potential to provide very large speedup, since their performance can scale up when the number of processors increases. This research effort evaluated how large scale scientific computational problems can be solved on two different distributed memory parallel systems at Carnegie Mellon - the Warp and Nectar systems.

Three applications were chosen. The first problem from the biological sciences and the second application involving chemical process modeling were implemented on Warp. The third application of representing 3-dimensional models was implemented on both Warp and Nectar.

2. The Warp and Nectar Systems

The Warp machine is a systolic computer of linearly connected cells, each of which is a programmable processor capable of performing 10 million operations per second (10 MFLOPS). A typical Warp array includes 10 cells, resulting in a peak computation rate of 100 MFLOPS. The Warp array can be extended to include more cells to accommodate applications capable of using the increased computational bandwidth. Warp is integrated as an attached processor into a UNIX host system. Programs for Warp are written in a high-level language, W2, supported by an optimizing compiler.

Nectar, which stands for NEtwork CompuTer ARchitecture, is a potentially larger system than Warp. In the Nectar System, processors, called nodes, are interconnected via a high-bandwidth, low-latency network consisting of fiber-optic links, a crossbar switch, and high-speed communication processors (Ref. 1). The nodes can be general purpose computers such as workstations, multiprocessor parallel machines such as Warp or high speed I/O devices. Due to its flexible and high-speed inter-node communication, Nectar has the potential to make effective use of a large number of powerful and heterogeneous nodes in one system.

3. Simulation of Hydrocarbon Monolayer

Computer simulations of complex macromolecules or supra-molecular assemblies of biological interest require a large amount of computing time. The difficulty increases when interactions or constraints with different energies are present simultaneously. Such a heterogeneous system typically exhibits dynamics on multiple time scales, and equilibrates over long periods of time. Parallel computers, which offer large processing power at a modest cost, are ideal for molecular simulations (Ref. 2).

In this work, we implemented a Monte Carlo (MC) simulation of a hydrocarbon monolayer implemented on the Warp systolic-array computer (Refs.3, 4). The hydrocarbon monolayer is considered as a simplified model for the hydrophobic region of a phospholipid bilayer, which is the structural component for biological membranes.

The model consisted of 40 hydrocarbon chains with 16 links in each chain. Each link consists of one carbon atom and one hydrogen atom. The monolayer is mapped onto the Warp architecture by dividing the entire system (40 chains) into 10 equally sized subsystems (4 chains). Each cell in the Warp array will then contain the coordinates and energy table for it's own 4 chains. At the beginning of every cycle, each cell picks one of it's chains at random and makes an MC move. The coordinates of each atom in this displaced chain are then calculated. This computation is independent of other cells and can be done in parallel across the Warp array. The coordinates of the displaced chain are then broadcast to all other cells. At this point, the non-bonding interaction energy is computed between the displaced chain and all other chains in the system. This computation between the displaced chain and the 4 chains which are allocated to that cell is performed locally and occurs in parallel across the Warp array. When this computation completes, all 10 cells communicate and a decision is made concerning whether or not the displaced chain will be accepted or not. If the displaced chain is accepted, each cell must update it's local energy table. Otherwise, the displaced chain is discarded and computation proceeds on another displaced chain.

The algorithm we have implemented exhibits a very high degree of parallelism and thus takes advantage of the Warp architecture. The Warp W2 program ran about 1/4 the speed of a well-vectorized FORTRAN program on a CRAY-XMP at the Pittsburgh Supercomputing Center. We estimate that the local memory limitation of 32Kbytes on Warp reduced its performance by a factor of two. Thus, the speed of the Warp is roughly equal to that of a CRAY-1. We conclude that the Warp systolic- array computer can successfully perform complex molecular simulations.

4. Chemical Process Modeling

Whether employed for simulation, design, optimization or control, the basic idea in process flow sheeting is to use mathematical models to describe the chemical dynamics occurring inside a plant. Typically, these models are large systems of nonlinear algebraic or differential equations. One of the most common techniques used in process flowsheeting to avoid the high computational cost associated with the evaluation of the partial derivatives is to use quasi-Newton methods. Instead of evaluating the true Jacobian matrix on each iteration, an update procedure is used to revise the Jacobian that was used in the previous iteration. Thus, in this work, particular attention was given to the use of the Warp array for implementing systolic algorithms for LU decomposition and back substitution.

The code for the computation was written in the AL language (Ref. 5)). AL is a compiler that generates all the necessary microcode for communication between the Warp processors. For a 15 cell Warp system, speed-ups of approximately ten over a single cell system were obtained for a range of problems. More specifically, performance levels exceeding 60 MFLOPS were achieved for a 184x184 system and greater than 90 MFLOPS for a 266x266 system. Such performance indicates that advanced parallel and distributed computer systems offer the potential to cut the time for large chemical process flowsheets by many orders of magnitude.

5. Three-Dimensional Models on Warp and Nectar

Octree decomposition (OD) is an algorithm used for representing 3-dimensional models. In such representations, a 3-dimensional object is approximated by a number of cubes of various sizes.

The octree is a data structure with a root and 8 possible successor nodes. Each of these successor nodes has in turn 8 or fewer successor nodes. This continues until a leaf node is reached; leaf nodes have no successor nodes. The octree is the same as a binary tree except each node can have up to 8 successor nodes instead of 2 as in the binary case. The conventional level terminology is as follows: the root node represents level 0; the eight successor nodes are called level -1 nodes and the highest level , leaves, are denoted by level-n nodes.

OD is used in many applications (Ref. 6) including solid modeling, 3-d finite mesh generation, computer graphics and computer vision. For all of these applications, OD imposes large computational and memory requirements making it an ideal candidate for implementing on Warp and Nectar.

The research addressed both the computation time and memory requirement problems on different fronts. The results are detailed in (Ref. 7). Here we will summarize the benchmark results of implementing OD on Nectar and Warp and , and for comparison, several other computer systems.

Figure 5-1 shows the benchmarks of the OD with Single-Point-In/Out-test on the highest level n. Note that the vertical axis is logarithmic.

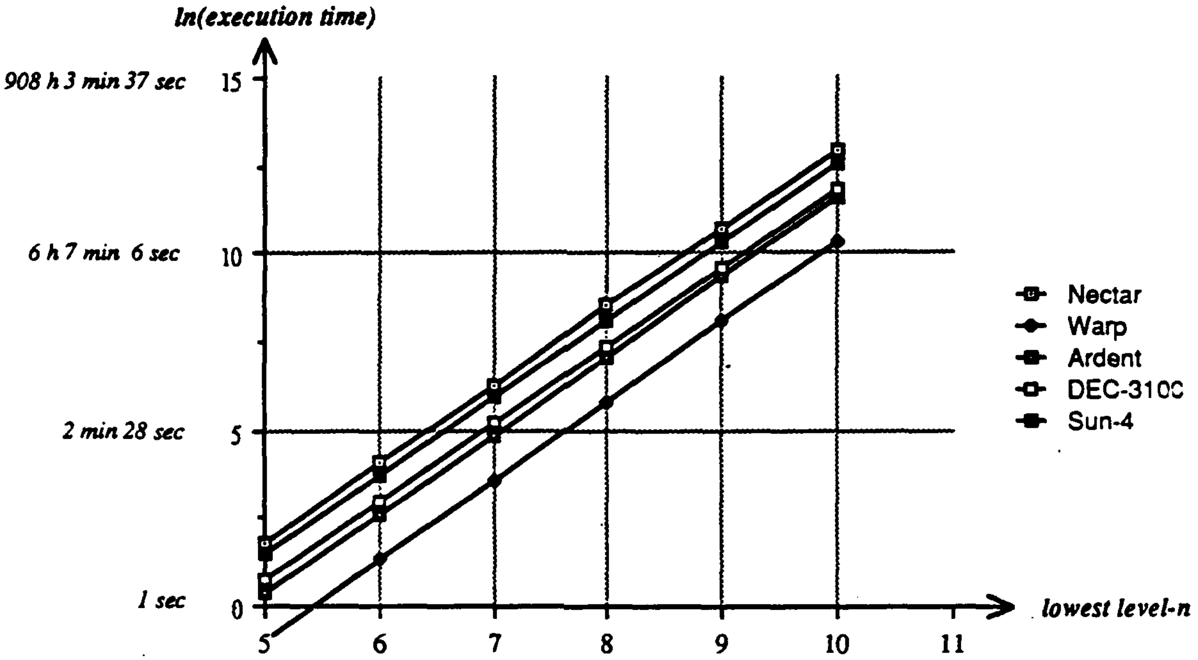

Figure 5-1: Octree Decomposition benchmarks on different computer systems

The execution time of each computer system shows a linear dependency relative to the highest level n. Theoretically, the execution time of level n+1 should be eight times that of level n. Measured times showed a factor of nine. The "inefficiencies" were due to operating system overhead, task switching and communication among processors.

It is interesting to note that only Warp approaches optimality. Warp has no operating system hence any task running simultaneously on Warp receives 100 % of the CPU time. Nectar's performance was hindered by the fact that the benchmarking was done with only two nodes, a Sun-3/60 and a Sun-3/50. However, other benchmarks on Nectar have shown that the the time of OD increases almost linearly with Nectar nodes. Hence, Nectar's relative performance would be improved significantly with the addition of nodes to the network.

6 Concluding Remarks

The three large scale scientific computational problems addressed in this effort clearly demonstrate that such problems can be effectively solved using distributed memory parallel machines such as the Warp and Nectar systems. Additional research is required in parallelizing such applications in order to take full advantage of distributed memory parallel systems.

References

1. Arnould, E. A., Bitz, F. J., Cooper, E. E., Kung, H. T., Sansom, R. D., Steenkiste, P. A., The Design of Nectar: A Network Backplane for Heterogeneous Multicomputers, CMU-CS-89-101, January , 1989.

2. Fincham, D., Parallel Computers and Molecular Simulation, Molecular Simulation 1, 1987, 1-45.

3. Kung, H. T., Why Systolic Architectures?, IEEE Computer Magazine, 15, 1982, 37-42

4. Annaratone, M., et al., The Warp Computer: Architecture, Implementation and Performance, IEEE Transactions on Computers, C-36(12), 1987, 1523-1538.

5. Tseng, P-S., A Parallelizing Compiler for Distributed Memory Machines, Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pa. 1989.

6. Kela, A., Hierarchical Octree Approximation for Boundary Representation based Geometric Models, June, 1988.

7. Fahringer, T., Prinz, F., B., Kung, H., T., New Methods of the Octree Decomposition, To be published.